

In the Claims:

1. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system, said lock system comprising:

a consumer lock, said consumer lock granted to one or more readers and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks ~~block~~ of data;

a producer lock, said producer lock granted to a single writer and said producer lock allowing a said writer granted said producer lock to ~~change~~ update said file comprising one or more blocks ~~block~~ of data, and

wherein upon completion of said ~~writer changing update~~ said block of data, said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with said reader granted said readers having a consumer lock associated with said updated file being ~~is notified~~ regarding said updatesaid block of data has changed.

2. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said ~~block of data file~~ is changed updated by writing ~~updated changed~~ blocks of data to a physical storage location different than where said block of data is stored.

3. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides

metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, ~~wherein said notification informs said reader of said updated data location wherein, after~~ said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file.

4. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per ~~claim 3~~ claim 2, wherein ~~said notification causes a cached copy of said block of data file is held~~ in a cache of said reader to be invalidated and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

5. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per ~~claim 3~~ claim 2, wherein reads performed on said block of data by said reader after receiving said notification are performed by reading said updated file data from at ~~said updated data-notified~~ location.

6. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein said reader continues to read said ~~block of data file~~ file from the physical storage

location ~~said block of data is stored~~ while said writer is writing ~~said~~ updated data to said different physical storage location.

7. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

8. (presently amended) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said writer writes data to said physically separate storage devices via that are part of a storage area network.

9. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said storage device located on said file system server stores metadata.

10. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said physically separate storage devices cache data for read operations.

11. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said reader is a web server.

12. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer is a database management system.

Claim 13 (cancelled)

14. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented on a system where said reader and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

15. (original) A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented in a distributed file system which utilizes multiple

locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data.

16. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a single writer to allow said writer to update said ~~data-block~~ file, said method comprising:

receiving a request from ~~said~~ a writer to grant ~~said-an exclusive~~ producer lock to ~~said~~ writer;

granting said producer lock to said writer;

receiving a producer lock release message, said producer lock release message being received after said writer completes updating said file ~~of said data-block~~; and

publishing said updated file and sending an update message to said readers holding said consumer lock, said update message notifying said readers ~~said data-block has been~~ updated regarding said update.

17. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a single writer to allow said writer to update said ~~data-block~~ file as per

claim 16, wherein said ~~data-block~~ file is updated by writing ~~updated~~ changed blocks of data to a different physical storage location than where said data block is stored.

18. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 17, wherein said update message informs said readers ~~of said updated data location~~ granted a consumer lock for said file regarding location of said updated file.

19. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per ~~claim~~ claim 17, wherein said update message causes a cached copy of said ~~data-block~~ file held in a cache of said readers ~~to be invalidated~~ and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

20. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 17, wherein reads performed on said data block by said readers after receiving said update

message are performed by reading said updated ~~data file~~ from at said updated data notified location.

21. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~file~~ data-block, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 17, wherein said reader continues to read said ~~data-block~~ file from the physical storage location ~~said data-block is stored~~ while said writer is writing said updated ~~data~~ file to said different physical storage location.

22. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~file~~ data-block, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 16, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

23. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~file~~ data-block, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 22, wherein said writer writes data to said physically separate storage devices ~~via~~ that are part of a storage area network.

24. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 22, wherein said storage device located on said file system server stores metadata.

25. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 22, wherein said physically separate storage devices cache data for read operations.

26. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 16, wherein said reader is a web server.

27. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 16, wherein said writer is a database management system.

28. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 16, wherein said lock system is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

29. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file ~~data-block~~, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 16, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data.

30. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said

producer lock granted to a writer to allow said writer to update said ~~data-block~~ file, said method comprising:

sending a request for said producer lock;
receiving said producer lock;
updating said ~~data-block~~ file comprising one or more data blocks;
releasing said producer lock after said updating is completed; and
publishing said updated file.

31. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 30, said method further comprising:

sending an update message to said readers granted said consumer lock after said releasing publishing_step, said update message notifying said readers said ~~data-block~~ file has been updated.

32. (presently amended) A method of updating a data block file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per ~~claim 30~~ claim 31, wherein said updating step comprises writing updated changed blocks of data to a different physical storage location than where said data block is stored.

Claim 33 (cancelled)

34. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per ~~claim-~~ 33 claim 31, wherein said notification update message informs said readers granted a consumer lock for said file regarding location of said updated file data location.

35. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per ~~claim-~~ 33 claim 32, wherein said ~~notification~~ update message causes a cached copy of said data block held in a cache of said readers to be invalidated and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

36. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per ~~claim-~~ 33 claim 34, wherein reads performed on said data block by said readers after receiving said ~~notification~~ update message are performed by reading said updated ~~data-file~~ data-notified location.

37. (presently amended) A method of updating a data-block-file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said data-block-file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data-block-file as per ~~claim~~ claim 32, wherein said readers continue to read said data-block file from the physical storage location ~~said-block-of-data-is-stored~~ while said writer is writing said-updated data to said different physical storage location.

38. (presently amended) A method of updating a data-block-file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said data-block file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data-block file as per claim 30, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

39. (presently amended) A method of updating a data-block-file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said data-block-file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data-block-file, as per claim 38, wherein said writer writes data to said physically separate storage devices via-that are part of a storage area network.

40. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~file as per claim 30, wherein said storage device located on said file system server stores metadata.

41. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~file as per claim 38, wherein said physically separate storage devices cache data for read operations.

42. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~file as per claim 30, wherein said reader is a web server.

43. (presently amended) A method of updating a ~~data-block~~file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~file as per claim 30, wherein said writer is a database management system.

44. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 30, wherein said method is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

45. (presently amended) A method of updating a ~~data-block~~ file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said ~~data-block~~ file, and a producer lock, said producer lock granted to a writer to allow said writer to update said ~~data-block~~ file as per claim 30, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data.

Claim 46 (cancelled)

Claim 47 (cancelled)

Claim 48 (cancelled)

Claim 49 (cancelled)

Claim 50 (cancelled)

Claim 51 (cancelled)

Claim 52 (cancelled)

Claim 53 (cancelled)

Claim 54 (cancelled)

Claim 55 (cancelled)

Claim 56 (cancelled)

Claim 57 (cancelled)

58. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocol, said system comprising:

a server, said server connected to at least one client of said distributed computing system via a first data network, said server serving file metadata to said client upon said client accessing a file stored in said distributed computing system, said server managing data consistency and cache coherency through ~~multiple said locking protocol protocols designed to support different application's data consistency and cache coherency needs;~~

a storage device connected to said client via a second data network, said storage device storing file data;

wherein one of said ~~multiple locking protocol protocols is assigned to a file based upon which application utilizes said file and said locking protocol assigned to said file is indicated via said file metadata~~ comprises the following locks:

a consumer lock, said consumer lock granted to one or more readers and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks of data; and

a producer lock, said producer lock granted to a single writer and said producer lock allowing said writer granted said producer lock to update said file comprising one or more blocks of data, and upon completion of said update, said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with readers having a consumer lock associated with said updated file being notified regarding said update.

Claim 59 (cancelled)

Claim 60 (cancelled)

61. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per ~~claim 60~~ claim 58, wherein said ~~block of data~~ file is changed by writing ~~updated~~ changed blocks of data to a physical storage location different than where said block of data is stored.

62. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per claim 61, wherein, after said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file ~~said notification informs said reader of said updated data location.~~

63. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols.

protocol, as per claim 62, wherein ~~said notification causes a~~ cached copy of said ~~block of data~~ file held in a cache of said reader ~~to be invalidated and changed blocks in said physical storage~~ are updated in said cached copy, thereby providing updates at a finer granularity.

64. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per ~~claim 61~~ claim 62, wherein reads performed on said ~~block of data by said reader~~ after receiving said notification file are performed by reading said updated data from said ~~updated data notified~~ location.

65. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per claim 61, wherein said reader continues to read said ~~block of data file~~ file from the physical storage location ~~said block of data is stored~~ while said writer is writing said updated ~~data file~~ data to said different physical storage location.

66. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per ~~claim 60~~ claim 58, wherein said reader is a web server.

67. (presently amended) A distributed computing system including a file system handling cache coherency and data consistency providing ~~multiple~~ quality of service through a locking protocols protocol, as per ~~claim 60~~ claim 58, wherein said writer is a database management system.

a1

Claim 68 (cancelled)
